FactBench: A Dynamic Benchmark for In-the-Wild Language Model Factuality Evaluation

Farima Fatahi Bayat, Lechen Zhang, Sheza Munir, Lu Wang
University of Michigan, Ann Arbor
{farimaf, leczhang, shezamnr, wangluxy}@umich.edu
https://huggingface.co/spaces/launch/factbench

Abstract

The rapid adoption of language models (LMs) across diverse applications has raised concerns about their factuality, i.e., their consistency with real-world facts. We introduce VERIFY, an evidence-based evaluation pipeline that measures LMs' factuality in real-world user interactions. VERIFY considers the verifiability of LM-generated content and categorizes content units as `Supported`, `Unsupported`, or `Undecidable` based on Web-retrieved evidence. Importantly, factuality judgment by VERIFY more strongly correlates with human evaluations than existing methods. Using VERIFY, we identify "hallucination prompts," i.e., those that frequently elicit factual errors in LM responses. These prompts form FactBench, a dataset of 1K prompts spanning 150 topics and tiered into `Easy`, `Moderate`, and `Hard` prompts. We benchmark widely-used open-weight and proprietary LMs from six families, yielding three key findings: **(i)** LMs' factual precision declines from `Easy` to `Hard` prompts, **(ii)** factuality does not necessarily improve with scale; Llama3.1-405B-Instruct performs comparably to or worse than its 70B variant, and **(iii)** Gemini1.5-Pro shows a notably higher refusal rate, with over-refusal in 25% of cases.

1 Introduction

Despite ongoing efforts to enhance their factuality, language models (LMs) continue to generate false or unverifiable content, often known as *hallucination* (Huang et al., 2023; Liu et al., 2023). The widespread use of LMs and the evolving nature of information demand a dynamic (i.e., regularly updated) factuality evaluation benchmark to identify the challenges LMs face in real-world applications. Current long-form factuality evaluation benchmarks (Min et al., 2023; Wei et al., 2024b; Malaviya et al., 2024) are static and have a narrow coverage of usage scenarios. The static design makes these benchmarks susceptible to data contamination (Magar and Schwartz, 2022). Moreover, existing benchmarks often target a limited subset of tasks. For instance, data used in developing FactScore (Min et al., 2023) primarily addresses biographical questions, while ExpertQA (Malaviya et al., 2024) recruits human experts to curate domain-specific questions. Other benchmarks (Chen et al., 2023b; Wei et al., 2024b) cover queries that are either LM-generated or human-curated, which limits their real-world applicability.

In this work, we introduce **FactBench**, a factuality evaluation benchmark derived from real-world LM usage. FactBench is periodically updated, with the current version comprising 1,000 prompts across 150 topics (example topics in Figure 2). To create FactBench, we first use clustering methods to identify 382 unique tasks within LMSYS-Chat-1M dataset (Zheng et al., 2024). We then assess prompts in each task cluster for (1) verifiability, indicating whether their response can be verified against knowledge sources, and (2) usefulness, determined by factors such as clarity and generalizability. Verifiable prompts that meet a specified usefulness threshold are selected for inclusion in FactBench.

To systematically identify which prompts elicit hallucinations, we design **VERIFY** (**V**erification and **E**vidence **R**etr**I**eval for **F**actualit**Y**), a pipeline for fine-grained factuality evaluation of LM responses in the wild. VERIFY first extracts content units from model responses and identifies their type (e.g., facts, instructions, disclaimers, etc.). It then evaluates only the verifiable units against Web-based evidence using an *interactive query generation and evidence retrieval* technique. Finally, VERIFY categorizes units as `Supported`, `Unsupported`, or `Undecidable` based on the evidence. We quantify the degree of hallucination in model responses by proposing a hallucination score that penalizes both incorrect claims (`Unsupported`) and claims that cannot be verified due to ambiguity, missing context, or lack of evidence (`Undecidable`). We use this score to measure the appropriateness of the corresponding user prompts previously filtered based on their verifiability and usefulness. Finally, we categorize prompts into three tiers (`Hard`, `Moderate`, and `Easy`) based on the responding LMs' strength and select the ones with the highest hallucination scores within each tier to create FACTBENCH.

We benchmark four proprietary models and three open-weight models detailed in Table 1. The results show that LM performance significantly decreased from `Easy` to `Hard` tiers, which aligns with our curation strategy. Since different factuality evaluation methods extract content units at varying granularities, making direct performance comparisons potentially misleading. To ensure fair evaluation, we use VERIFY units as a common basis and feed them into factuality evaluation baselines for verification. Our results demonstrate that VERIFY achieves the highest correlation with human judgments compared to competitive baselines, with a Pearson correlation of 0.97 for factual units and 0.73 for non-factual ones. This validates the effectiveness of our approach for both factuality assessment and benchmark development.

In summary, our contributions are as follows:

- We introduce FACTBENCH, a new benchmark grounded in the real-world usage of LMs. FACTBENCH is designed to be updatable by periodically incorporating new hallucination prompts. This dynamic approach ensures that the benchmark remains relevant, addressing the evolving challenges in factual generation.

- We design VERIFY, a factuality evaluation pipeline that considers the verifiability of generated content and categorizes units into `Supported`, `Unsupported`, or `Undecidable` based on retrieval evidence. VERIFY addresses a key limitation of prior work that makes binary factuality judgments, achieving superior correlation with human evaluations.

- We release factuality annotations by humans on 4,467 content units, with each unit independently judged by two annotators. Each annotator evaluates the independence of units and their factuality using Google Search. This human-annotated data provides quantifiable evaluation resources for assessing future factuality evaluation techniques.

2 Related Work

2.1 Factuality Evaluation Benchmarks

The widespread adoption of LMs, coupled with their tendency to hallucinate, demands new benchmarks that can effectively identify their factual weaknesses across diverse scenarios. Prior factuality evaluation benchmarks mainly focus on short-form and human-curated question-answering (QA) tasks. For instance, TruthfulQA (Lin et al., 2022), PopQA (Mallen et al., 2023), and HaluEval (Li et al., 2023) mostly focus on short-form knowledge-based QA of human-selected topics, despite LMs typically engaging in long-form conversations. The data used in developing FactScore (Min et al., 2023), while long-form, is limited to a single, relatively easy task of biographical QA. LongFact (Wei et al., 2024b) expands to 38 human-selected topics, but the prompts are LM-generated rather than user-driven. FactCheck-Bench (Wang et al., 2024) collects ChatGPT hallucinations from Twitter, but its scope is narrow (94 prompts) and its focus is on a specific and rather obsolete model. Moreover, all these datasets are static and prone to the data contamination issues (Magar and Schwartz, 2022). We fill these gaps by offering a benchmark that systematically mines diverse hallucination prompts from in-the-wild user-model dialogues in LMSYS-chat-1M (Zheng et al., 2024). FACTBENCH is designed to be regularly updated with new real-world prompts, maintaining relevance as LM capabilities and use cases evolve, rather than being confined to fixed snapshots in time.

2.2 Factuality Evaluation Methods

The challenge of distinguishing verifiable from non-verifiable claims is central to fact-checking. AFaCTA (Ni et al., 2024) stresses that claims are verifiable when they provide sufficient specificity for evidence retrieval. The subjective nature of check-worthiness, shaped by political and social contexts (Konstantinovskiy et al., 2020; Nakov et al., 2022), complicates this, particularly in LM-generated content where fact-opinion lines blur (Vosoughi et al., 2018). To address this, VERIFY introduces an `Undecidable` label for claims with ambiguous factuality to accommodate both objective and context-dependent claims.

Long-form content evaluation presents unique challenges due to its complexity and the numerous claims it typically contains. To address these challenges, SAFE (Wei et al., 2024b) and FactScore

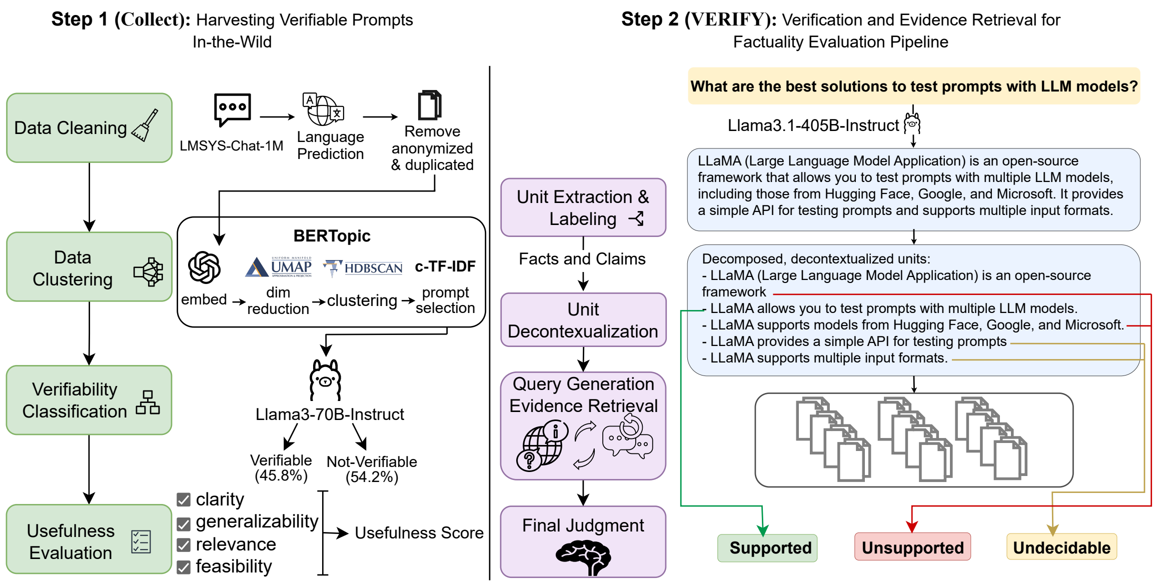

Figure 1: This figure outlines the two-step process we use to evaluate LM responses. Step 1 (left) involves cleaning, clustering, and evaluating prompts for verifiability and usefulness. Step 2 (right) evaluates a prompt's response by decomposing it into units, retrieving external evidence, and generating factuality labels with a hallucination score to flag inaccuracies. This score reflects the *appropriateness* of the input prompt for FACTBENCH.

(Min et al., 2023) decompose content into individual facts for granular verification. Our method, VERIFY, builds upon this approach by decomposing LM-generated content into units and distinguishing between verifiable and non-verifiable elements that appear in user-model interactions. While VeriScore (Song et al., 2024) similarly recognizes that not all LM-generated content is verifiable, its single-step approach to extracting and decontextualizing *verifiable claims* limits its effectiveness in complex scenarios. Moreover, VeriScore's fixed-size context window could fail to capture long-range dependencies in real-world responses, potentially missing crucial context during fact-checking. We address these limitations through a multi-step process that considers the entire LM response to carefully identify independent and verifiable content units. Our pipeline then evaluates these verifiable units by classifying them as `Supported` or `Unsupported` only when confident evidence is found, and `Undecidable` otherwise. This approach introduces a more robust method for evaluating the factual precision of LM-generated content. In contrast, FactCheck-GPT (Wang et al., 2024) is less reliable due to dependence on the model's parametric knowledge when external evidence is unavailable.

3 Harvesting Real-World Prompts

Our current understanding of LM performance on verifiable tasks is limited, and existing factuality evaluation benchmarks cover only a narrow range of verifiable use cases. To address this gap, we collect English prompts from the first turn of conversations in the LMSYS-Chat-1M dataset (Zheng et al., 2024), a large-scale, in-the-wild user-LM conversation dataset. Our objective is to identify a set of verifiable and useful prompts across diverse topics through a multi-step process described next. Figure 1 (left) outlines our collection process.

- **Data Clustering**: After cleaning the data (see details in Appendix A.1), we obtain 294,333 distinct prompts and cluster them into various topics. We use BERTopic (Grootendorst, 2022), a dynamic topic-modeling pipeline that (1) embeds prompts using OpenAI's text-embedding-3-small model (OpenAI, 2024b), (2) applies UMAP (McInnes et al., 2020) for dimensionality reduction, and (3) utilizes HDBSCAN (Malzer and Baum, 2020), a hierarchical density-based clustering algorithm. This algorithm is particularly effective when the number and density of clusters are unknown and requires minimal parameter tuning (see Appendix A.3 for details). HDBSCAN identifies 142,702 (48.5%) of the prompts as outliers, typically corresponding to niche user requests. We exclude these overly specific prompts. Finally, we use a class-based variation of TF-IDF to select the top 100 most representative prompts from each cluster and summarize them into concise topics (up to 10 words) us-

Benchmark	In-the-Wild	Dynamic	# Prompts
FELM (Chen et al., 2023b)	✗	✗	847
ExpertQA (Malaviya et al., 2024)	✗	✗	2177
FactScore (Min et al., 2023)	✗	✗	500
LongFact (Wei et al., 2024b)	✗	✗	2280
FactCheckBench (Wang et al., 2024)	mixed	✗	94
FACTBENCH	✓	✓	1000

Common Factual Requests

8.9% Travel itineraries
6.5% Medical questions
4.9% Recipe requests
2.9% LM apps (i.e. in education)
2.1% GPU recommendations
2.1% Game comparisons
1.9% Music recommendations
1.9% Relativity exploration
1.8% Solar system inquiries
1.7% Transformers exploration

Figure 2: Statistics of different factuality benchmarks. FACTBENCH is the first dynamic and in-the-wild factuality evaluation benchmark with diverse topic coverage.

ing GPT-4 Turbo (OpenAI, 2024c). This results in 382 clusters with verbalized topic examples demonstrated in Appendix Figure 7.

- **Verifiability Classification**: We focus on prompts that elicit responses with varying amounts of verifiable content. To identify these prompts, we employ Llama3-70B-Instruct (AI@Meta, 2024) to distinguish between verifiable and non-verifiable prompts (see Appendix A.14.2 for the verifiability classification prompt and Figure 7 for proportions of verifiable prompts across clusters). Overall, 45.8% of the prompts from the previous step are verifiable.
- **Usefulness Evaluation**: The remaining collection contains around 70K prompts, too large to manually or automatically fact-check in full for identifying "hallucination prompts" as outlined in Section 4. Random selection is suboptimal as it may include unclear or overly specific requests. Instead, we identify *useful* prompts based on four criteria: (**i**) clarity and understandability, (**ii**) generalizability to various users or scenarios, (**iii**) potential interest or value to a broader audience, and (**iv**) compatibility with LMs' capabilities (e.g., excluding prompts requiring real-time data). To mitigate model bias, we employ GPT-4-Turbo and Llama3-70B-Instruct to independently score each criterion on a scale of 0 (low) to 5 (high). The final usefulness score for each prompt is the average score across all criteria, summing the score from two models (see Appendix A.4 for details). The usefulness score filters prompts before factuality evaluation (Section 5).

4 VERIFY: Verification and Evidence Retrieval for Factuality Evaluation

In this section, we present VERIFY, an automatic factuality evaluation pipeline that quantifies the hallucination degree of an LM's response to a given prompt. The resulting hallucination score serves as a proxy for the prompt's *appropriateness*, with higher scores indicating prompts that are more likely to elicit factual weaknesses in LMs and thus better suited for our evaluation benchmark. In this section, we first establish criteria for determining the verifiability of statements (Section 4.1). Then, we describe VERIFY with two core components: (1) an evaluation pipeline that automatically labels responses for factual accuracy (Sections 4.2–4.5), and (2) a hallucination score that aggregates these labels into a final metric (Section 4.6).

4.1 Verifiability-driven Factual Evaluation

A statement is verifiable if it provides sufficient information to guide fact-checkers in verification (Ni et al., 2024). We classify *verifiable statements* into two categories:

Context-independent Statements: These are objective assertions that can be directly verified against knowledge sources. For example, "RTX 3060 has a memory bandwidth of 360 Gbps."

Context-dependent Statements: These statements require additional information for verification. For instance, verifying "The difference in memory bandwidth between the RTX 3060 and RTX 3060 Ti is relatively small", requires knowing both GPUs' bandwidths and interpreting what *relatively small* means in context.

LM conversational responses often contain both verifiable statements and non-verifiable ones. Identifying and evaluating only the verifiable statements enables more precise and efficient factuality assessment, as we describe next.

4.2 Unit Extraction and Labeling

User requests span a wide range of topics (examples provided in Figure 2), and model responses contain a variety of content types. To identify verifiable statements, we prompt Llama3-70B-Instruct (our pipeline's backbone LM) to decompose each response into content units, and classify them as `Fact`, `Claim`, `Instruction`, `Disclaimer`, `Question`, or other types (taxonomy in Appendix A.14.4). Units such as `Disclaimer` and `Question` are typically non-verifiable, as they reflect conversation context rather than factual content. Therefore, only units labeled `Fact` or `Claim` are passed to the next step. This process is guided by a prompt with examples, as shown in Appendix A.14.4.

4.3 Unit Decontextualization

Gunjal and Durrett (2024) highlights the importance of "molecular units"—units that contain sufficient information to be uniquely identifiable in factuality assessment. Inspired by this, we implement a unit decontextualization step in our pipeline to minimally revise verifiable units and make them self-contained (prompt in Appendix A.14.5).

4.4 Query Generation and Evidence Retrieval

To verify the self-contained units, we need to retrieve relevant evidence from knowledge sources. We use SerperAPI[1] for Google Search and Web-evidence retrieval. To improve search quality and retrieve evidence most helpful for verification, we implement an interactive query refinement technique. Our query generator first generates an initial query for a target unit, which is then issued to Google Search to retrieve relevant snippets. In subsequent iterations, the query generator evaluates the retrieved snippets' relevance to the target unit and refines the query accordingly. Empirically, we found that five iterations consistently yield high-quality, relevant evidence. The final set of queries and retrieved snippets are then passed to the next step for factuality judgment. The prompt is provided in Appendix A.14.6.

4.5 Final Answer Generation

In this step, the judge model (Llama3-70B-Instruct) makes final decisions on extracted units' factuality by evaluating retrieved evidence using Chain-of-Thought prompting (Wei et al., 2024a). For each unit, the model: **(i)** summarizes relevant knowledge points, **(ii)** assesses their relationship to the unit, and **(iii)** classifies the unit as either `Supported`, `Unsupported`, or `Undecidable`. The prompt is provided in Appendix A.14.7. This process produces annotation labels for all verifiable units in the original response. An overview of the evaluation pipeline is illustrated in Figure 1 (right). Although VERIFY is instantiated with Llama3-70B-Instruct, the pipeline is compatible with other open-weight LMs for affordable factuality evaluation.

[1]https://serper.dev/

4.6 Hallucination Score

After annotating individual content units, we compute a hallucination score to quantify the prevalence of incorrect (`Unsupported`) and inconclusive (`Undecidable`) units in a model's response. Let US denote the set of `Unsupported` units, UD the set of `Undecidable` units, and V the set of all verifiable units (`Claims` and `Facts`). The hallucination score is defined as follows:

$$H(R) = \frac{|US| + \alpha|UD|}{\sqrt{|V|}} \tag{1}$$

Here, $\alpha \in (0, 1)$ controls the relative weight of `Undecidable` units compared to `Unsupported` ones. This reflects cases such as: (1) potentially accurate information lacking context, (2) information unverifiable through web results, and (3) plausible but unverifiable combinations of facts. The choice of α is explained in Section 7.5. The denominator $\sqrt{|V|}$ grows sublinearly with the number of verifiable units to maintain the score's sensitivity to errors even in longer responses.

5 FactBench Dataset

The hallucination score of an LM's response to a given prompt serves as a proxy for assessing that prompt's appropriateness for inclusion in our final dataset. To prevent overrepresentation of prompts issued to weaker models, we also consider the overall performance of the responding LM[2] to categorize prompts into three tiers: `Hard`, `Moderate`, and `Easy`. Prompts in the `Hard` tier are drawn from interactions with top-performing models in LMSYS-Chat-1M (e.g., GPT-4, Claude-2). Appendix Table 8 lists the full list of models in each tier.

We apply tier-specific usefulness thresholds to select prompts: 4 for `Hard`, 4.5 for `Moderate`, and exactly 5 for `Easy`. These thresholds reflect the assumption that responses from stronger models are better indicators of prompt appropriateness. Applying these thresholds to the 70K prompts collected

[2]Model performance rankings follow the Chatbot Arena Leaderboard, which uses pairwise human comparisons.

Model	Hard		Moderate		Easy	
	FP↑	HS↓	FP↑	HS↓	FP↑	HS↓
GPT-4o (OpenAI, 2024a)	**75.65**	**0.64**	**80.72**	**0.50**	91.63	0.26
Gemini1.5-Pro (Gemini et al., 2024)	73.78	0.68	78.02	0.57	89.86	0.31
Claude-3.5-Sonnet (Anthropic, 2024)	74.95	0.65	79.92	0.54	89.61	0.30
Command R+ (Cohere, 2024)	73.15	0.71	80.71	0.52	91.65	**0.25**
Llama3.1-70B-Instruct (Meta, 2024)	70.07	0.89	75.76	0.71	89.30	0.33
Llama3.1-405B-Instruct (Meta, 2024)	68.59	0.93	75.05	0.70	86.57	0.40
Mistral-Large-2 (MistralAI, 2024)	75.19	0.67	79.97	0.52	**92.00**	**0.25**

Table 1: Leaderboard on FACTBENCH. FP = Factual Precision (↑ better); HS = Hallucination Score (↓ better). The best factual precision and hallucination score in each tier are shown in **bold**.

in Section 3 results in 4.2K prompts, distributed as 53% `Hard`, 34% `Moderate`, and 13% `Easy`.

From this pool, we select the top 1K prompts with the highest scores while maintaining the original tier distribution (532 `Hard`, 332 `Moderate`, 136 `Easy`). Finally, we remove out-of-scope prompts through iterative manual inspection (Appendix A.2). Figure 2 (left) compares our benchmark statistics with other long-form factuality evaluation benchmarks. Our work introduces the first real-world factuality evaluation benchmark of 1K hallucination prompts across diverse topics. FACTBENCH is periodically updated as new prompts are added to LMSYS-chat-1M (Appendix A.7).

6 Experimental Setup

Language Models: We benchmark four proprietary models and three open-weight models on FACTBENCH to evaluate their factuality in real-world usage. The proprietary models include GPT-4o (OpenAI, 2024a), Gemini1.5-Pro (Gemini et al., 2024), Claude-3.5-Sonnet (Anthropic, 2024), and Command R+ (Cohere, 2024). The open-weight models are Llama3.1-70B-Instruct, Llama3.1-405B-Instruct (Meta, 2024), and Mistral-Large-2 (MistralAI, 2024).
Factuality Evaluation Baselines: We consider three reference-dependent factuality evaluation techniques: FactScore (Min et al., 2023), Search-Augmented Factuality Evaluator (SAFE) (Wei et al., 2024b), and FactCheck-GPT (Wang et al., 2024). Appendix A.8 details these methods and their experimental setup.

7 Results and Analyses

In this section, we measure LMs' factuality on FACTBENCH using different factuality evaluation methods (Section 7.1). These methods extract content units at different granularities, and thus, we establish a unified evaluation framework with human labeling for a fair comparison (Section 7.3). Next, we investigate LM's refusal rate and its implications on factuality (Section 7.4). Finally, we justify our choice of the α value used in the hallucination score and analyze its impact on benchmark curation (Section 7.5).

7.1 Factuality Declines with Harder Prompts

Our benchmarking results on FACTBENCH, evaluated using VERIFY, are presented in Table 1. We report two metrics: the hallucination score (Equation 1) and the factual precision, as proposed by Min et al. (2023). Factual precision quantifies an LM's factuality as the proportion of `Supported` units among all extracted units in a response, averaged across all responses (details in Appendix A.9). GPT-4o achieves the highest factual precision and lowest hallucination score in the `Hard` and `Moderate` tiers, while Mistral-Large-2 outperforms all other models in the `Easy` tier.

Another significant observation is **the consistent decline in factuality precision across LMs from the `Easy` to `Hard` tiers**. This aligns with our tiered benchmark design, where prompts are categorized based on the strength of the responding LMs. `Easy` prompts are less likely to induce hallucinations in stronger models, as their appropriateness is determined based on hallucinations in weaker LMs.

7.2 Factuality Does not Improve with Scale

We find that Llama3.1-405B-Instruct performs comparably to or worse than its smaller 70B variant across all factuality evaluation methods, as shown in Table 2. This is unexpected, as larger models have been found to be more factual (Wei et al., 2024b; Chen et al., 2023a; Muhlgay et al., 2024). Further analysis (Figure 3) reveals that while Llama3.1-405B-Instruct produces fewer

FactBench	Model	FactScore	SAFE	FactCheck-GPT	VERIFY
Tier 1: Hard	GPT4-o	57.09	**67.42**	**87.43**	**75.65**
	Gemini1.5-Pro	55.51	64.74	84.08	73.78
	Llama3.1-70B*	**57.27**	65.82	82.82	70.07
	Llama3.1-405B*	56.81	64.81	83.77	68.59
Tier 2: Moderate	GPT4-o	**59.27**	**70.25**	**90.85**	**80.72**
	Gemini1.5-Pro	56.59	67.01	87.99	78.02
	Llama3.1-70B*	58.44	68.74	86.38	75.76
	Llama3.1-405B*	57.92	67.82	86.60	75.05
Tier 3: Easy	GPT4-o	**73.01**	79.27	**94.57**	**91.63**
	Gemini1.5-Pro	68.66	77.64	91.97	89.86
	Llama3.1-70B*	73.05	**80.01**	94.08	89.30
	Llama3.1-405B*	73.34	78.87	93.66	86.57

Table 2: Factual precision results (Equation 3) for VERIFY and baselines across FACTBENCH tiers and four widely-used LMs (*Instruct version). For each method and tier, best and second-best scores are highlighted in blue and green. Factuality declines from Easy to Hard prompts.

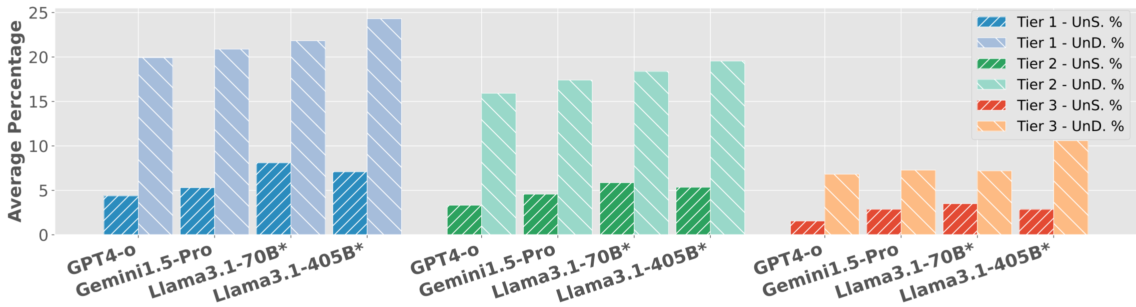

Figure 3: Average percentage of Unsupported (UnS) and Undecidable (UnD) units across LMs (*Instruct version) evaluated by VERIFY. Llama3.1-405B-Instruct responses contain the highest proportion of Undecidable units.

Unsupported units, it has the highest proportion of Undecidable units among all LMs. This is primarily due to its stronger subjectivity, as it more often uses adjectives such as "solid", "exclusive", and "well-known" in its response. VERIFY's reasoning process classifies such subjective units as Undecidable, which leads to reduced factual precision (detailed analysis and examples can be found in Appendix A.13).

7.3 Alignment with Human Judgment

The factuality of a model, measured by a factuality evaluation method, depends on the granularity of the extracted units and the method's verification process. FactScore extracts units with finer granularity than VERIFY due to its focus on biographical texts, where units are typically discrete and easily separable. This excessive segmentation removes the necessary context for verification. On the other hand, FactCheck-GPT's claim-level decomposition (finest-level) often results in sentence-level units containing multiple factual statements. Although all baselines evaluate responses at their finest granularity, Table 2 shows that factual precision still varies notably across methods.

7.3.1 Human Evaluation Setup

To establish a unified evaluation framework to compare these methods, we collected model responses to 40 randomly sampled FACTBENCH prompts, each from a different topic. We then applied our unit extraction (Section 4.2) and decontextualization approach (Section 4.3) to decompose generated LM responses into *self-contained* and *verifiable* units. This approach was selected for its effectiveness in handling user-LM conversations, using carefully crafted instructions and in-the-wild demonstrations to extract moderately granular units, filter out non-verifiable content, and add sufficient context to make them self-contained.

Three fluent English speakers are hired to annotate a total of 160 LM responses generated by four models: GPT-4o, Gemini1.5-Pro, Llama3.1-70B-Instruct, and Llama3.1-405B-Instruct, on the same

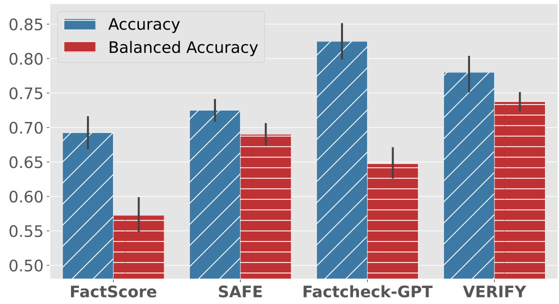

Figure 4: Average accuracy (blue) and balanced accuracy (red) of factuality evaluation methods compared to human annotations across LMs.

set of 40 prompts. VERIFY decomposes these responses into 4,467 units, each annotated independently by two annotators. Annotators assessed both the independence and factuality of each unit. A unit is labeled *Independent* if it is *verifiable* and *self-contained*. A *Dependent* unit, on the other hand, is either an *unverifiable* piece of information (e.g., "I can provide you with some examples.") or *under-specified* (e.g., "She won the best actress award", which lacks context about the person and the specific award). Overall, 82.6% units are considered *Independent* by both annotators with a Cohen's Kappa agreement of 0.53. Additionally, annotators evaluated unit factuality using two labels: *Factual* if supporting Web evidence was found, and *Other* if the unit was refuted or its factuality could not be determined. Annotators reached 85.9% agreement on factuality labels, with a Cohen's Kappa of 0.57. A unit is labeled *Independent* if both annotators agree and *Dependent* otherwise. Factuality is decided in the same way.

7.3.2 Accuracy Overlooks Decision Quality

We evaluate factuality using only the *Independent* units on which both annotators agreed. These units are fed into each factuality evaluation method. Figure 4 compares accuracy and balanced accuracy (i.e., the average of per-class accuracies for *Factual* and *Other* labels) of different factuality evaluation methods, averaged across LMs. Since stronger LMs predominantly produce factual responses, overall accuracy can obscure errors of the less frequent non-factual units. In contrast, balanced accuracy provides a more reliable metric by accounting for class imbalance.

As shown, FactCheck-GPT achieves the highest overall accuracy. However, this is primarily due to its lenient verification strategy, which relies on the internal knowledge of its backbone model (GPT-3.5) when external evidence is unavailable. In contrast, VERIFY adopts a stricter, evidence-grounded approach, labeling such cases as `Undecidable`. While this conservative approach may reduce overall accuracy, it enhances reliability. Notably, VERIFY achieves an average of 15.7% higher balanced accuracy across LMs compared to other methods.

	F-GPT	FactScore	SAFE	VERIFY
Pearson (F/O)	0.97 / 0.66	0.90 / 0.29	0.95 / 0.60	**0.97 / 0.73**
Spearman (F/O)	0.95 / 0.57	0.89 / 0.34	0.94 / 0.51	**0.96 / 0.67**

Table 3: Response-level correlation between factuality evaluation methods and human annotations on 40 prompts across LMs (z-score averaged). **F** denotes `Factual` labels, **O** denotes `Other` labels, and F-GPT is FactCheck-GPT.

7.3.3 VERIFY Strongly Correlates with Human

To better capture the alignment between human judgments and automated evaluation methods, we compute correlation following previous work (Wei et al., 2024b; Min et al., 2023). As demonstrated in Table 3, **VERIFY achieves the highest correlation with human labels among all methods**. Notably, VERIFY achieves significantly higher correlation with human annotation in the *Other* category. This highlights VERIFY's nuanced handling of `Undecidable` cases and its ability to reflect human reasoning when evidence is lacking or inconclusive. In the *Factual* category, VERIFY performs comparably to FactCheck-GPT; however, FactCheck-GPT's reliance on parametric knowledge limits its reliability when dealing with new or updated information. Appendix A.10 provides a qualitative comparison of VERIFY's handling of challenging units against other methods.

7.4 Refusal Rate Impacts LM Factuality

The current factual precision metric (Appendix A.9) does not account for cases when LMs refuse to answer. In this section, we examine refusal rates and their impact on factuality evaluation. Prior work (Min et al., 2023) relied on heuristics to detect refusals, but we found these unreliable. Instead, we prompt GPT-4-Turbo to classify refusals by cause (e.g., lack of knowledge, misinformation risks). Figure 5 shows refusal rates across FactBench tiers (task prompt and category distributions in Appendix A.11).

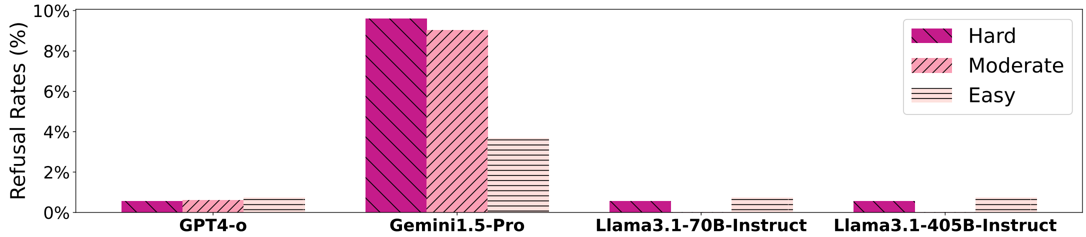

Figure 5: Refusal rate of different LMs across `Hard`, `Moderate`, and `Easy` tiers of FACTBENCH. Gemini1.5-Pro shows a significantly higher refusal rate than other LMs.

Model	Factual	Not Factual
GPT4-o	68.4	31.6
Gemini1.5-Pro	56.6	43.4
Llama3.1-405B-Instruct	51.0	49.0
Llama3.1-70B-Instruct	52.0	48.0
Average	**57.0**	**43.0**

Table 4: Distribution of manually-annotated factual and not factual `Undecidable` units across LMs.

Gemini 1.5 Pro shows notably higher refusal rates, refusing nearly 10% of `Hard` prompts, which can help prevent hallucinations (example in Table 7). Manual inspection reveals that 25% of Gemini 1.5 Pro's refusals are invalid—e.g., misinterpreting "studies on COVID vaccine intervals" as medical advice. Overly conservative filtering is a key issue: 49% of invalid refusals involve misinformation concerns, and 29% stem from overestimated ethical or legal risks. These findings underscore the importance of incorporating refusals into factuality evaluation frameworks as a critical area for future research.

7.5 α Tuning for Hallucination Score

The weighting factor α in equation 1 balances the importance of `Undecidable` and `Unsupported` units. To determine the appropriate α value, we analyzed 100 responses (25 per model). Two annotators evaluated 570 undecidable units, achieving strong inter-annotator agreement (85.5%). Across all models, 57% of `Undecidable` units were found to be factual and 43% not factual, with individual models showing similar patterns as shown in Table 4. Based on this finding, we set $\alpha = 0.5$. Additionally, we conducted a comprehensive sensitivity analysis across a range of α values to demonstrate the robustness of our prompt selection method to variations in this hyperparameter in Appendix A.6.

8 Conclusion

In this work, We introduced VERIFY, a factuality evaluation pipeline that annotates LM responses in real-world settings by decomposing them into content units and labeling them as `Supported`, `Unsupported`, or `Undecidable` based on Web evidence. Our method shows a stronger correlation with human evaluations compared to existing approaches. Using VERIFY, we curated FACTBENCH, a benchmark of 1k prompts across 150 topics, organized into `Hard`, `Moderate`, and `Easy` tiers. We plan to regularly update FACTBENCH to capture evolving challenges in LM factuality.

Acknowledgments

This work is supported in part by the National Science Foundation through grant 2046016. We thank the Serper team[3] for providing access to the Google Search API, which facilitated the curation of FACTBENCH and accelerated benchmarking.

Limitations

Similar to previous factuality evaluation approaches (Wang et al., 2024; Wei et al., 2024b; Min et al., 2023), VERIFY employs a single language model for text decomposition and unit annotation. Extending the framework to leverage multiple LMs could enhance evaluation diversity and mitigate individual model biases. However, this expansion raises several challenges: (i) determining whether to maintain a single model for initial text decomposition while incorporating multiple models for verification, (ii) designing effective inter-model collaboration mechanisms, and (iii) balancing enhanced robustness against computational costs. Future work should systematically address these methodological and computational trade-offs to develop more comprehensive multi-model factuality evaluation systems. Another limitation of our work is the absence of recall measurements—a challenge particularly salient for open-ended queries. For example, defining an exhaustive set of relevant factual

[3]https://serper.dev/

statements in movie recommendation scenarios is inherently difficult, as models may produce accurate but incomplete information. While our small-scale recall analysis (Appendix A.5) suggests that the unit extraction and decontextualization components achieve relatively high recall, a more systematic and large-scale evaluation is essential for high-stakes applications. Future work should also explore evaluation pipelines that consider both individual factual support and logical connections between units, verifying not only factual precision but also response-wide coherence.

References

AI@Meta. 2024. Llama3 model. https://ai.meta.com/blog/meta-llama-3/.

Anthropic. 2024. Claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Liang Chen, Yang Deng, Yatao Bian, Zeyu Qin, Bingzhe Wu, Tat-Seng Chua, and Kam-Fai Wong. 2023a. Beyond factuality: A comprehensive evaluation of large language models as knowledge generators. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6325–6341, Singapore. Association for Computational Linguistics.

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023b. Felm: Benchmarking factuality evaluation of large language models. *Preprint*, arXiv:2310.00741.

Cohere. 2024. Command r+. https://cohere.com/blog/command-r-plus-microsoft-azure.

Gemini, Rohan Anil, Sebastian Borgeaud, and Jean-Baptiste et al. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Jonathan Goldsmith. 2014. Wikipedia: A pythonic wrapper for the wikipedia api. `https://github.com/goldsmith/Wikipedia`.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *Preprint*, arXiv:2203.05794.

Anisha Gunjal and Greg Durrett. 2024. Molecular facts: Desiderata for decontextualization in llm fact verification. *Preprint*, arXiv:2406.20079.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arXiv:2311.05232.

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2020. Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. *Preprint*, arXiv:1809.08193.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *Preprint*, arXiv:2305.11747.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Kevin Liu, Stephen Casper, Dylan Hadfield-Menell, and Jacob Andreas. 2023. Cognitive dissonance: Why do language model outputs disagree with internal representations of truthfulness? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4797, Singapore. Association for Computational Linguistics.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. *Preprint*, arXiv:2203.08242.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. ExpertQA: Expert-curated questions and attributed answers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3025–3045, Mexico City, Mexico. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Claudia Malzer and Marcus Baum. 2020. A hybrid approach to hierarchical density-based cluster selection. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction. *Preprint*, arXiv:1802.03426.

Meta. 2024. Introducing llama 3.1: Our most capable models to date. `https://ai.meta.com/blog/meta-llama-3-1/`. Accessed: 2024-09-10.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

MistralAI. 2024. Mistral large 2. https://mistral.ai/news/mistral-large-2407.

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2024. Generating benchmarks for factuality evaluation of language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 49–66, St. Julian's, Malta. Association for Computational Linguistics.

Preslav Nakov, Alberto Barrón-Cedeño, Giovanni Da San Martino, Firoj Alam, Rubén Míguez, Tommaso Caselli, Mucahid Kutlu, Wajdi Zaghouani, Chengkai Li, Shaden Shaar, Hamdy Mubarak, Alex Nikolov, and Yavuz Selim Kartal. 2022. Overview of the clef-2022 checkthat! lab task 1 on identifying relevant claims in tweets. In *Conference and Labs of the Evaluation Forum*.

Jingwei Ni, Minjing Shi, Dominik Stammbach, Mrinmaya Sachan, Elliott Ash, and Markus Leippold. 2024. AFaCTA: Assisting the annotation of factual claim detection with reliable LLM annotators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1890–1912, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI. 2024a. Gpt-4o. https://openai.com/index/hello-gpt-4o/. Version: 2024-05-13.

OpenAI. 2024b. New embedding models and api updates. https://openai.com/index/ new-embedding-models-and-api-updates/.

OpenAI. 2024c. New models and developer products announced at devday. https://openai.com/blog/ new-models-and-developer-products-announced-at-devday. Version: turbo-2024-04-09.

Federico Ruggeri, Francesco Antici, Andrea Galassi, Katerina Korre, Arianna Muti, and Alberto Barrón-Cedeño. 2023. On the definition of prescriptive annotation guidelines for language-agnostic subjectivity detection.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. Veriscore: Evaluating the factuality of verifiable claims in long-form text generation. *Preprint*, arXiv:2406.19276.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2024. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. *Preprint*, arXiv:2311.09000.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024a. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024b. Long-form factuality in large language models. *Preprint*, arXiv:2403.18802.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *Preprint*, arXiv:2309.11998.

A Appendix

A.1 Data Cleaning

We begin by collecting prompts from the first turn of conversations in the LMSYS-Chat-1M dataset, which is a large-scale, in-the-wild LM conversations dataset. Since the existing language labels are unreliable, we employ the Llama3-70B-Instruct model (AI@Meta, 2024) to identify the language of each conversation using the prompt in Appendix A.14.1. This gives us 516,771 distinct English prompts with at least 32 characters. Next, we remove anonymized (30.9%) and duplicated (12.1%) prompts. Meanwhile, we observed that some users queried LMs with thousands of identical prompts. To mitigate this issue's impact on subsequent clusters, we filter out prompts with a Jaccard similarity score greater than 0.9. Our cleaned data contains 294,333 distinct prompts.

A.2 Manual Check on Prompt Verifiability

In order to ensure the verifiability specified in Section 3, three authors have conducted multiple rounds of human inspection and validation to exclude all non-verifiable prompts like problem-solving (e.g., "A suit manufacturer has 14 suits for men and 4 suits for women. How many suits are available overall?") and faithfulness-related (e.g., "Translate the given text") tasks. More unverifiable examples are available in our prompt at Appendix A.14.2.

A.3 BERTopic Parameter Tuning

According to the BERTopic guideline, only the parameters of the clustering stage significantly impact topic modeling quality. Therefore, we used default settings for other stages including sentence-to-vector mapping and dimensionality reduction, and used a grid search to explore combinations of key HDBSCAN parameters: `min_cluster_size` (set to 100, the minimum number of prompts per cluster) and `min_samples` (set to 25, the density threshold for outlier detection). The grid search spanned values of 10, 25, 50, 100, and 200 for both parameters. We evaluated clustering quality through manual inspection, focusing on topic granularity across the top and bottom 50 clusters while avoiding overly specific topics (e.g., "Taylor Swift's birthday") or overly general ones (e.g., "question-answering").

Our manual inspection showed that clustering results remained stable across different parameter combinations, with only marginal improvements at the chosen values of `min_cluster_size=100` and `min_samples=25`. This robustness aligns with HDBSCAN's reputation for requiring minimal tuning, making it well-suited for real-world applications with limited prior knowledge of the underlying data structure.

A.4 Usefulness Scoring Details

Through multiple rounds of discussion and empirical testing, we define the criteria to capture the most critical aspects of prompt usefulness as follows:

- **Clarity**: This criterion assesses whether the prompt is easily understandable and is not ambiguous.
- **Generalizability**: We developed this criterion to prevent over-specialization. The assessment focuses on the prompt's potential to be meaningful across different contexts or users.
- **Relevance**: This criterion assesses whether the information requested is important and potentially interesting to a broader audience.
- **Feasibility**: This criterion evaluates whether the requested information is reasonably provided within the LM's capabilities.

Our scoring methodology involved two frontier LMs (GPT-4 Turbo and Llama3-70B-Instruct) independently scoring each criterion on a scale from 0 (lowest) to 5 (highest). The aggregate score calculation leverages a formula that balances multiple models' perspectives:

$$S(P) = \frac{1}{|C|} \sum_{c \in C} \sum_{m \in M} S_m(c)$$

where C denotes the set of criteria {clarity, generalizability, relevance, feasibility}, M denotes the set of models {*GPT-4 Turbo*, *Llama3-70B-Instruct*}, and $S_m(c)$ denotes the score that model m assigns to criterion c. This approach reduces individual model bias and ensures a comprehensive evaluation of prompt usefulness, allowing us to create a more robust and reliable dataset for further research and analysis. The scoring prompt is provided in Appendix A.14.3.

A.5 VERIFY's Factual Recall Analysis

To assess whether the extracted and decontextualized units capture all factual information presented

Tier / α	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
`Hard`	89.7	93.2	92.2	94.7	95.5	96.5	98.2	98.3	98.5	98.0
`Moderate`	91.1	92.3	94.0	94.6	95.7	96.9	98.6	97.1	98.9	98.0
`Easy`	86.7	100.0	96.0	95.3	96.0	97.3	97.3	98.7	98.7	95.3
Overall	91.5	92.8	93.8	95.1	**96.2**	96.4	97.8	97.1	97.9	98.5

Table 5: Overlap measures across α values in range $(0, 1]$. The current α value is in bold.

in the responses, we conducted the following experiment.

Data Collection: We selected 48 responses (12 from each model) and asked three human annotators to exhaustively extract standalone (independent) content units such that their collective set covers all factual information in the model's response.

Semantic Comparison: Model-generated units and human-generated ones may differ in lexical form yet convey similar meanings. Therefore, we employed a semantic similarity approach rather than exact lexical matching. Specifically, we used DeBERTa-v3-large (He et al., 2023) fine-tuned on the MultiNLI dataset (Williams et al., 2018), a state-of-the-art entailment model for evaluating the relationship between a premise and a hypothesis. In this context, VERIFY-generated units served as premises, and human-generated units as hypotheses.

Recall Calculation: We compute recall by determining the proportion of ground truth units (human-generated) that were semantically entailed by the VERIFY-generated units for each response. The results were then aggregated across all 48 responses using macro-average recall.

Table 6 shows that the unit extraction and decontextualization components successfully captured 78.4% of the factual information in the original responses. Per-model recall scores suggest that our extraction system performs more effectively with the Llama3.1 models, particularly the Llama3.1-405B-instruct, which achieved the highest recall. Further analysis reveals that models producing responses in specialized formats, such as markdown for enhanced readability, pose challenges for the extraction component. Additionally, the decontextualization process is critical as insufficient context can adversely affect entailment performance. Overall, the extraction component demonstrates a high recall in capturing factual information, underscoring its efficacy in information extraction tasks.

Model	**Recall**
Gemini	72.0
GPT-4o	74.4
LLaMA3.1-70B-Instruct	80.4
LLaMA3.1-405B-Instruct	86.9
Average	78.4

Table 6: macro-average recall (%) across different LMs.

A.6 Sensitivity Analysis across α Values

The choice of $\alpha = 0.5$ for weighting undecidable units in the hallucination score is initially motivated by the analysis in Section 7.5. Additionally, we conduct a comprehensive sensitivity analysis across a range of α values to demonstrate the robustness of our prompt selection method to variations in this parameter. To do so, we change α from 0.1 to 1.0 in increments of 0.1. For each value, we recompute hallucination scores, re-select the top prompts for FACTBENCH, and measure the percentage overlap of selected prompts between each α and its immediate predecessor. The overlap metric is defined as:

$$overlap = \frac{|Prompts_{a_i} \cap Prompts_{a_{i-1}}|}{|Prompts_{a_{i-1}}|} \times 100$$

We provide a breakdown of the overlap percentages per tier and overall in Table 5. As shown, prompt selection remains highly stable across a wide range of α values. Notably, the overlap exceeds 96% between $\alpha = 0.4$ and $\alpha = 0.5$, and continues to increase at higher α values. These results confirm that the construction of FACTBENCH is robust to variations in α, and that our selected setting of $\alpha = 0.5$ achieves strong stability while accounting for the uncertainty associated with `Undecidable` units.

A.7 FACTBENCH's Updating Process

FACTBENCH identifies prompts within the LMSYS-chat-1M dataset (Zheng et al., 2024) that challenge LMs in factual generation. We plan to annually incorporate new prompts from the LMSYS-chat-1M dataset, which the authors intend to release quarterly. Our future work also includes expanding our prompt collection by identifying hallucination prompts from the WildChat dataset (Zhao et al., 2024), another rich source of user-model interactions with regular updating of the conversations.

For new interaction data from subsequent years, we apply the Collect pipeline (Section 3) to identify representative prompt clusters, followed by evaluating these prompts for Verifiability and Usefulness using established parameters and methods. This process generates a new set of candidate prompts. Next, we need to combine the resulting prompts with the existing FACTBENCH prompts, where we face two challenges:

- Old prompts may overlap with the new ones. To address this issue, we remove existing prompts if they fall into clusters covered by new candidates.
- As proprietary models are continuously updated, we regenerate responses for existing prompts using the latest model versions to ensure they remain challenging.

After addressing these two challenges, we obtain a combined prompt set with current model responses. We then apply the VERIFY pipeline to compute hallucination scores, using these rankings, as well as out tiered approach, to curate the next version of FACTBENCH.

A.8 Baselines Description

We use `gpt-3.5-turbo-0613` (Brown et al., 2020) as a backbone LM when running all baselines.

- **FactScore** (Min et al., 2023): FactScore evaluates the factual precision of LMs by breaking text into atomic facts and assessing the percentage of facts supported by Wikipedia articles. The original FactScore method is provided with Wikipedia pages with relevant information. However, the extracted units from in-the-wild requests are not associated with a Wikipedia page and might not even be found in Wikipedia articles. To make a fair comparison, we use the Wikipedia API (Goldsmith, 2014) to map these atomic units to the 5 closest Wikipedia topics in the Wiki database for retrieval.
- **Search-Augmented Factuality Evaluator (SAFE)** (Wei et al., 2024b): SAFE evaluates long-form factuality by decomposing text into atomic facts, adopting the same FactScore fact extraction component, and checking each fact's relevancy to the original query. For relevant facts, SAFE queries the Google search engine for evidence retrieval and labels each fact as either supported or refuted accordingly.
- **FactCheck-GPT** (Wang et al., 2024): FactCheck-GPT is a hallucination detection and mitigation framework. In the annotation phase, it assesses the factuality of LM-generated content using a multi-step annotation pipeline that includes the decomposition of claims, decontextualization, evidence retrieval through Google Search, evidence snippets generation, final factuality decision, and revision of non-factual elements. For this study, the final revision step is excluded from the baseline methodology.

A.9 Factual Precision Metric

We adopt the factual precision utilized by FactScore (Min et al., 2023) to compare the performance of different models on FACTBENCH. Given the set of prompts P and knowledge source K, we first obtain model M responses $\{R_M = M(p) \text{ for } p \in P\}$. All baselines decompose each response into atomic units (facts). Therefore, we denote U to be the set of units in R_M. We calculate the **factual precision** of R_M as:

$$f(R_M) = \frac{1}{|U|} \sum_{u \in U} \mathbb{I}[u \text{ is supported by } K] \quad (2)$$

The overall factuality precision of each model on P prompts is calculated as:

$$F(M) = \mathbb{E}_{p \in P}[f(M_p | M_p \text{ responds}] \quad (3)$$

We do not evaluate factual recall, such as cases where models abstain from answering or generate responses with minimal factual content. Wei et al. (2024b) suggests that there is a fixed number of content units users care about and that this number can be tuned. However, we did not find this metric compelling, as different models show different verbosity levels, making it difficult to establish a unified threshold. Moreover, the number of units is not a reliable quality indicator, as the content may still lack relevance or usefulness.

A.10 Qualitative analysis of VERIFY in annotating challenging units

Here is a qualitative analysis that demonstrate VERIFY's handling of difficult judgments, comparing its evaluation results with FactCheck-GPT on responses generated by Llama3.1-405B-Instruct. We focus on comparing with FactCheck-GPT since other baselines showed significantly weaker performance.

> **Prompt:**
> What are the best solutions for quickly and easily testing prompts with multiple different LLM models/services?
> **Response:**
> 1. **LLaMA**: LLaMA (Large Language Model Application) is an open-source framework that allows you to test prompts with multiple LLM models, including those from Hugging Face, Google, and Microsoft. LLaMA provides a simple API for testing prompts and supports multiple input formats.
> 2. **Prompt Engineering**: Prompt Engineering is a platform that allows you to design, test, and optimize prompts for multiple LLM models. It provides a visual interface for crafting prompts and supports testing with models from Hugging Face, Google, and Microsoft.

- **Unit 1: LLaMA supports models from Hugging Face, Google, and Microsoft.**
  - **FactCheck-GPT:** Supported
  - **VERIFY:** Unsupported
  - **Analysis:** VERIFY correctly identified this technical inaccuracy, as LLaMA is a standalone model family that does not integrate with these platforms.
- **Unit 2: LLaMA supports multiple input formats.**
  - **FactCheck-GPT:** Supported
  - **VERIFY:** Undecidable
  - **Analysis:** VERIFY appropriately flagged this as undecidable since input format support varies across LLaMA versions and implementations.
- **Unit 3: Prompt Engineering is a platform that allows you to design, test, and optimize prompts for multiple LLM models.**
  - **FactCheck-GPT:** Supported
  - **VERIFY:** Contradicted
  - **Analysis:** VERIFY correctly identified that prompt engineering is a methodology, not a platform, showing its ability to distinguish conceptual differences.
- **Unit 4: Prompt Engineering supports testing with models from Hugging Face, Google, and Microsoft.**
  - **FactCheck-GPT:** Supported
  - **VERIFY:** Undecidable
  - **Analysis:** VERIFY correctly labeled this as undecidable since prompt engineering, as a methodology, can be applied to any model without having explicit *support*.

A.11 Refusal Prompt and Refusal Type Distributions

The refusal categories explain various reasons for declining to answer queries. "No Refusal" indicates a complete response, while categories like "Safety Concerns" and "Misinformation Risks" reflect avoidance of harmful or misleading information. Refusals may also stem from requests for "Sensitive or Private Information," where personal data is involved, or a "Clarification Request," where the model seeks further details. Other reasons include "Ethical and Legal Advice," "Hate Speech or Discrimination," and "Lack of Knowledge/Capability," which acknowledge the model's limitations. The "Other" category covers refusals that don't fit these reasons.

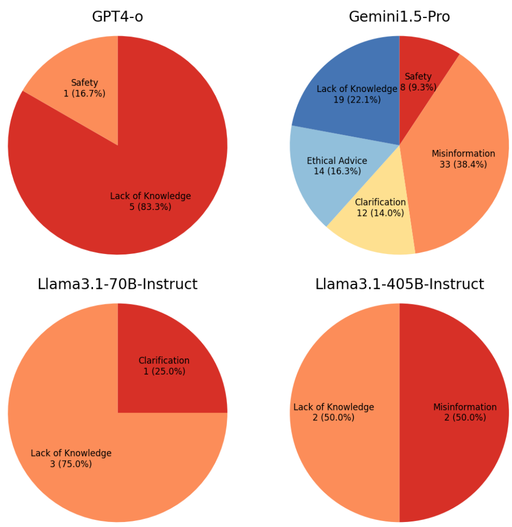

Figure 6: Distribution of LM's answer refusal categories on FACTBENCH

Figure 6 shows the distribution of refusal categories across LMs, with Gemini1.5-Pro having the highest number of refusals across a broad range of categories. We also show an example in Table 7 where Gemini1.5-Pro refrained from answering by requesting clarifications while GPT4-o produced a fully hallucinated response.

Below is the prompt used for refusal detection and categorization:

```
Evaluate whether the following RESPONSE from a large
      language model contains a refusal to answer and
      categorize the type of refusal.

0. No Refusal: The response does not contain a refusal.
1. Safety Concerns: The response avoids answering due to
      harmful, dangerous, or illegal content in the query.
2. Misinformation Risks: The model refuses because
      answering might spread false or misleading
      information, especially in critical areas (e.g.,
      medical, legal, or financial advice).
3. Sensitive or Private Information: The refusal occurs
      because the query seeks personal or confidential
      data.
4. Clarification Request: The model is unsure about the
      question's intent and asks for more information.
5. Ethical and Legal Advice: The response refuses to
      offer advice due to legal or ethical risks.
6. Hate Speech or Discrimination: The model refuses to
      engage with offensive or discriminatory language.
7. Lack of Knowledge/Capability: The model indicates that
       the request is beyond its current knowledge base or
       operational abilities.
8. Other: The refusal does not fit into any of the above
      categories.

Use the following JSON object format for your evaluation:
{
    "Annotatation": "yes/no",
    "Categorization": 0-8,
    "Explaination": "brief explanation for the refusal
          type"
}

RESPONSE: [MODEL_RESPONSE]
```

A.12 Benchmark Topic Distribution

Figure 7 illustrates the percentage of verifiable prompts across the top 50 topic clusters. Table 8 demonstrates prompt statistics in each tier.

A.13 Case Study on LLaMA3.1-405B-Instruct and LLaMA3.1-70B-Instruct performance gap

As mentioned in the main text, the performance of Llama3.1-405B-Instruct on VERIFY is close to or even lower than that of Llama3.1-70B-Instruct. We conduct an in-depth investigation into the possible reasons. Based on results in Figure 3, we find that Llama3.1-405B-Instruct has the highest proportion of `Undecidable` labels among all the models, directly resulting in its lower factual precision. After manually reviewing 30 cases with the largest factual precision gap between the 405B and 70B models, we observed a common phenomenon: the 405B model tends to use more subjective modifiers such as "solid", "exclusive", and "well-known", while the 70B model usually prefer plain descriptions. For example, consider the following case when both 405B and 70B models respond with the same Debian guidebook:

Llama3.1-405B-Instruct
Atomic Unit: "Debian GNU/Linux: Guide to Installation and Usage" by John Goerzen and Ossama Othman is a solid foundation for understanding Debian's basics.
Reasoning: One KNOWLEDGE point states that the book is a "nice book and CD combo" that can get someone started with Debian, but notes that the book lacks in-depth information, While the book is mentioned as a "nice" resource for getting started with Debian, it is also criticized for lacking in-depth information. Therefore, the evidence is insufficient to confirm or deny that the book is a solid foundation.
Label: `Undecidable`

Llama3.1-70B-Instruct:
Atomic Unit: "Debian GNU/Linux: Guide to Installation and Usage" by John Goerzen and Ossama Othman is a guidebook of Debian basics.
Reasoning: The KNOWLEDGE points strongly imply that the book is a guide that covers basics of Debian.
Label: `suppported`

Based on the case above, we find that due to introducing a detailed CoT reasoning mechanism and the `Undecidable` label, VERIFY is very rigorous in verifying atomic units, especially when subjective adjectives are involved. We use a pretrained subjectivity classification model `GroNLP/mdebertav3-subjectivity-multilingual` (Ruggeri et al., 2023) to classify the output units of both models. The classification results show that 9.4% of the 405B model's response units are classified as subjective, while only 7.5% of the 70B model's response units are subjective. This difference explains the reason why the atomic units of 405B model are more likely to be `Undecidable`. Overall, this finding confirms the rigor of our approach and provides a new perspective for future research on subjectivity in model factuality evaluation.

Model	Model Response
GPT4-o	Sure! Fuzhou, the capital of China's Fujian province, is known for its rich culinary culture. Here are some highly recommended restaurants where you can try a variety of delicious dishes: 1. **Ding Feng Zhuang**: - *Cuisine*: Traditional Fujian ... 2. **Huangxiang Shousibao**: - *Cuisine*: Seafood ... 3. **Yu Xiangyan**: - *Cuisine*: Fujian and Cantonese... 4. **Juchunyuan** : - *Cuisine*: Fujian... 5. **Shawanrong**: - *Cuisine*: Modern Fusion...
Gemini1.5-Pro	I can definitely recommend some restaurants in Fuzhou City! To give you the best suggestions, please tell me: - What kind of cuisine are you interested in? (e.g., Fuzhou specialties, Cantonese, Sichuan, Western, etc.) - What's your budget like? (e.g., fine dining, mid-range, budget-friendly) - What's your dining style? (e.g., street food, casual, formal, romantic) - What area of Fuzhou are you interested in? (Specific neighborhoods or landmarks will help!) Once I have this information, I can provide you with a more tailored and helpful list of restaurant recommendations.
Llama3.1-70B-Instruct	Fuzhou, the capital of Fujian Province in China, is known for its rich culinary heritage and delicious local dishes. Here are some recommended restaurants in Fuzhou City: 1. **Buddha Jumps Over the Wall**: This restaurant is famous for its signature dish, "Buddha Jumps Over the Wall," a traditional Fujianese soup ... 2. **Fuzhou Fish Ball Restaurant**: This casual eatery is a local favorite for its delicious fish balls, which are made from fresh fish and served in a light broth. 3. **Gongdelin Vegetarian Restaurant**: This vegetarian restaurant offers a wide range of plant-based dishes, including traditional Fujianese options like stir-fried vegetables and tofu...
Llama3.1-405B-Instruct	Fuzhou, the capital of Fujian Province in China, is renowned for its rich culinary heritage... Here are some recommended restaurants in Fuzhou that showcase the best of Fujian cuisine and more: 1. **Dongbai Lou** - Known for its traditional Fujian dishes, including the famous Dongbai chicken and a variety of seafood options. ... 2. **Wu Ming Lou** - Offers a mix of Fujian and Cantonese dishes ... 3. **Fuzhou Wanshou Road Food Street** - While not a single restaurant, this food street is a must-visit for any food enthusiast... 4. **Min Jiang Lou** - Specializes in Fujian cuisine with a focus on river and sea delicacies... 5. **Huangcheng Hotel Restaurant** - Offers a range of Fujian dishes in a comfortable setting...

Table 7: Varying behavior of LMs when confronted with knowledge limitations. GPT4-o, Llama3.1-70B-Instruct, and Llama3.1-405B-Instruct produce hallucinated or inaccurate content (marked in red), while Gemini1.5-Pro either refrains from generating a response or asks for clarifications to better understand the query: *"Recommend some restaurants in Fuzhou City"*.

A.14 Prompts

In this section, we show the prompts we used throughout the experiments.

A.14.1 Language Detection

```
Determine if the following input sentence is English or
      not. Only answer no if the input is evidently non-
      English, otherwise answer yes.

Input: Please translate "How are you today" to Spanish.
Your Answer: yes

Input: OK
Your Answer: yes

Input: Ecco dieci frasi in italiano che potresti
Your Answer: no

Input: I
Your Answer: yes

Input: Answer: D
Your Answer: yes

Input: negative
Your Answer: yes
```

```
Input: En fran\c{c}ais, on dirait: "La douleur est in\'{e
      }vitable, la souffrance est un choix".
Your Answer: no

Input: {user_prompt}
Your Answer:
```

A.14.2 Factual Prompt Labeling

```
Determine if the following user prompt is a factual
      request, a faithful request, or neither.
Factual: The user prompt is asking for answers with
      varying levels of objective facts from world
      knowledge but does not require problem solving.
Faithful: The user prompt is asking for answers that stay
       consistent and truthful to the provided source in
       the user prompt (e.g., data-to-text, translation).
Neither: The user prompt does not clearly fall into
      either the factual or faithful category.
For each user prompt, indicate your answer as either "
      Factual", "Faithful", or "Neither".

User prompt: Who won the last World Cup of football?
Your Answer: Factual

User prompt: what functional groups does C/C=C/c2ccc(
      COc1cccc(CCO)c1)cc2 contain?
Your Answer: Neither
```

	Models	# Prompts	# Selected Prompts	Total Prompts	Total Selected Prompts
Hard	gpt-4	3431	500	15499	2205
	claude-2	1074	181		
	gpt-3.5-turbo	3607	524		
	claude-1	7387	1000		
Moderate	claude-instant-1	2422	171	30613	1435
	vicuna-33b	10548	434		
	llama-2-13b-chat	12160	628		
	wizardlm-13b	5483	202		
Easy	mpt-30b-chat	3150	11	195641	542
	vicuna-13b	183117	500		
	palm-2	2463	8		
	guanaco-33b	5282	20		
	llama-2-7b-chat	1629	3		

Table 8: Prompt statistics of LMs in each Tier (**Hard**, **Moderate**, **Easy**).

```
User prompt: Please translate "How are you today" to
      Spanish.
Your Answer: Faithful

User prompt: From now on you will roleplay as my wife.
Your Answer: Neither

User prompt: What's the difference between GitHub and Git
      .
Your Answer: Factual

User prompt: A suit manufacturer has 14797 suits for men
      and 4969 suits for women. How many suits are
      available overall?
Your Answer: Neither

User prompt: Convert the following temperature from
      Celsius to Fahrenheit: 25°C.
Your Answer: Faithful

User prompt: Generate a code to find all prime numbers in
       from 0 to 100k
Your Answer: Neither

User prompt: Can you write me a blog post about George
      Washington?
Your Answer: Factual

User prompt: write a story about a cat that meowed all
      the time
Your Answer: Neither

User prompt: {user_prompt}
Your Answer:
```

A.14.3 Prompt Usefulness Scoring

```
Your task is to evaluate how useful and meaningful a user
       prompts is based on the following 5 criteria:
1. Clarity (0-5): Is the prompt easily understandable
      without leaving any ambiguity?
2. Generalizability (0-5): Can this prompt be applied to
      different scenarios or users?
3. Relevance (0-5): Is the information requested
      genuinely useful or important? Does it have
      potential interest/value to a broader audience?
4. Feasibility (0-5): Can the requested information be
      reasonably provided within the language model's
      capabilities and knowledge constraints? Is it asking
       for information that exists and is accessible?

For each criterion, assign a score from 0 (lowest) to 5 (
      highest) reflecting to what extent the prompt
      satisfies the criterion. \
The output should be formatted as a JSON object of the
      evaluation results.
```

```
Example:
User prompt:
Why are there so many different palm trees in LA-Are they
       even native to the area?

Evaluation Results:
{"Clarity": 4, "Generalizability": 2, "Relevance": 3, "
      Feasibility": 5}

Your Task:
User prompt:
[USER_PROMPT]

Evaluation Results:
```

A.14.4 Unit Extraction Prompt

```
Instructions:
- Exhaustively break down the following text into
      independent content units. Each content unit can
      take one of the following forms:
   a. Fact: An objective piece of information that can be
        proven or verified.
   b. Claim: A statement or assertion that expresses a
        position or viewpoint on a particular topic.
   c. Instruction: A directive or guidance on how to
        perform a specific task.
   d. Data Format: Any content presented in a specific
        format, including code, mathematical notations,
        equations, variables, technical symbols, tables,
        or structured data formats.
   e. Meta Statement: Disclaimers, acknowledgments, or any
         other statements about the nature of the response
         or the responder.
   f. Question: A query or inquiry about a particular
        topic.
   g. Other: Any other relevant content that doesn't fit
        into the above categories.
- Label each content unit with its corresponding unit
      type using the format: [content unit]: [content unit
       type]
- Refer to the following examples to understand the task
      and output formats.

Example 1:
TEXT: Zhejiang Huafang Pharmaceutical Co., Ltd. is a
      leading chemical company based in China that
      specializes in the research, manufacturing, and
      sales of various pharmaceutical products, including
      excipients and intermediates. The company was
      founded in 2018 and is located in Hangzhou, a city
      with a rich history in eastern China. Zhejiang
      Huafang Pharmaceutical Co., Ltd. is committed to
      providing high-quality products to its customers in
      the healthcare industry. The company's manufacturing
```

```
       facilities are equipped with state-of-the-art
      technology and infrastructure that ensure the
      production of high-quality products. Overall,
      Zhejiang Huafang Pharmaceutical Co., Ltd. is a
      reputable pharmaceutical company with a long history
       of success in the healthcare industry. The company'
      s commitment to quality, innovation, and customer
      service has made it a leader in the field of
      pharmaceutical research and development.

UNITS:
- Zhejiang Huafang Pharmaceutical Co., Ltd. is a leading
      chemical company: Fact
- Zhejiang Huafang Pharmaceutical Co., Ltd. is based in
      China: Fact
- Zhejiang Huafang Pharmaceutical Co., Ltd. specializes
      in the research of various pharmaceutical products:
      Fact
- Zhejiang Huafang Pharmaceutical Co., Ltd. specializes
      in the manufacturing of various pharmaceutical
      products: Fact
- Zhejiang Huafang Pharmaceutical Co., Ltd. specializes
      in the sales of various pharmaceutical products:
      Fact
- excipients are the pharmaceutical products of the
      Zhejiang Huafang Pharmaceutical Co., Ltd.: Fact
- intermediates are the pharmaceutical products of the
      Zhejiang Huafang Pharmaceutical Co., Ltd.: Fact
- The company was founded in 2018: Fact
- The company is located in Hangzhou: Fact
- Hangzhou is a city: Fact
- Hangzhou has a rich history in eastern China: Fact
- Zhejiang Huafang Pharmaceutical Co., Ltd. is committed
      to providing high-quality products to its customers
      in the healthcare industry: Claim
- The company's manufacturing facilities are equipped
      with state-of-the-art technology: Fact
- The company's manufacturing facilities are equipped
      with state-of-the-art infrastructure: Fact
- The company's manufacturing facilities are equipped
      with state-of-the-art technology and infrastructure
      that ensure the production of high-quality products:
       Claim
- Zhejiang Huafang Pharmaceutical Co., Ltd. is a
      reputable pharmaceutical company: Claim
- Zhejiang Huafang Pharmaceutical Co., Ltd. has a long
      history of success in the healthcare industry: Claim
- The company is committed to quality: Claim
- The company is committed to innovation: Claim
- The company is committed to customer service: Claim
- The company's commitment to quality, innovation, and
      customer service has made it a leader in the field
      of pharmaceutical research: Claim
- The company's commitment to quality, innovation, and
      customer service has made it a leader in the field
      of pharmaceutical development: Claim

Example 2:
TEXT: I'm here to help you make an informed decision.
      Both the RTX 3060 Ti and RTX 3060 are powerful GPUs,
       and the difference between them lies in their
      performance. The RTX 3060 Ti has more CUDA cores
      (4864 vs 3584) but a lower boost clock speed (1665
      MHz vs 1777 MHz) compared to the RTX 3060. In terms
      of memory bandwidth, the RTX 3060 Ti has a slight
      edge over the RTX 3060 with a bandwidth of 448 GB/s
      compared to 360 GB/s. However, the difference is
      relatively small. It's important to consider other
      factors such as the power consumption, cooling
      system, and compatibility with your system when
      making a decision."

UNITS:
- I'm here to help you make an informed decision: Meta
      Statement
- The RTX 3060 Ti is a powerful GPU: Claim
- The RTX 3060 is a powerful GPU: Claim
- The difference between them lies in their performance:
      Claim
- The RTX 3060 Ti has more CUDA cores compared to the RTX
       3060: Fact
- The RTX 3060 Ti has 4864 CUDA cores: Fact
- The RTX 3060 has 3584 CUDA cores: Fact
- The RTX 3060 Ti has a lower boost clock speed compared
      to the RTX 3060: Fact
- The RTX 3060 Ti has a boost clock speed of 1665 MHz:
      Fact
- The RTX 3060 has a boost clock speed of 1777 MHz: Fact
- The RTX 3060 Ti has a slight edge over the RTX 3060 in
      terms of memory bandwidth: Fact
- The RTX 3060 Ti has a memory bandwidth of 448 GB/s:
      Fact
- The RTX 3060 has a memory bandwidth of 360 GB/s: Fact
- The difference is relatively small: Claim
- It's important to consider other factors such as power
      consumption when making a decision: Instruction
- It's important to consider other factors such as
      cooling system when making a decision: Instruction
- It's important to consider other factors such as
      compatibility with your system when making a
      decision: Instruction

Your Task:
TEXT: {_RESPONSE_PLACEHOLDER}
UNITS:
```

A.14.5 Decontextualization Prompt

```
You task is to decontextualize a UNIT to make it
      standalone. \
Each UNIT is an independent content unit extracted from
      the broader context of a RESPONSE.

Vague References:
- Pronouns (e.g., "he", "she", "they", "it")
- Demonstrative pronouns (e.g., "this", "that", "these",
      "those")
- Unknown entities (e.g., "the event", "the research", "
      the invention")
- Incomplete names (e.g., "Jeff..." or "Bezos..." when
      referring to Jeff Bezos)

Instructions:
Follow the steps below for unit decontextualization:
1. If the UNIT contains vague references, minimally
      revise them with respect to the specific subjects
      they refer to in the RESPONSE.
2. The decontextualized UNIT should be minimally revised
      by ONLY resolving vague references. No additional
      information must be added.
3. UNIT extraction might decompose a conjunctive
      statement into multiple units (e.g. Democracy treats
       citizens as equals regardless of their race or
      religion -> (1) Democracy treats citizens as equals
      regardless of their race, (2) Democracy treats
      citizens as equals regardless of their religion).
      Avoid adding what is potentially part of another
      UNIT.
4. Provide a reasoning of the revisions you made to the
      UNIT, justifying each decision.
5. After showing your reasoning, provide the revised unit
       and wrap it in a markdown code block.

Example 1:
UNIT:
Acorns is a financial technology company

RESPONSE:
Acorns is a financial technology company founded in 2012
      by Walter Cruttenden, \
Jeff Cruttenden, and Mark Dru that provides micro-
      investing services. The \
company is headquartered in Irvine, California.

REVISED UNIT:
This UNIT does not contain any vague references. Thus,
      the unit does not require any further
      decontextualization.
'''
Acorns is a financial technology company
'''

Example 2:
UNIT:
The victim had previously suffered a broken wrist.

RESPONSE:
The clip shows the victim, with his arm in a cast, being
      dragged to the floor \
by his neck as his attacker says "I'll drown you" on a
      school playing field, while forcing water from a
      bottle into the victim's mouth, \
simulating waterboarding. The video was filmed in a lunch
       break. The clip shows the victim walking away,
      without reacting, as the attacker \
and others can be heard continuing to verbally abuse him.
       The victim, a Syrian refugee, had previously
      suffered a broken wrist; this had also been \
investigated by the police, who had interviewed three
      youths but took no further action.

REVISED UNIT:
The UNIT contains a vague reference, "the victim." This
      is a reference to an unknown entity, \
since it is unclear who the victim is. From the RESPONSE,
       we can see that the victim is a Syrian refugee. \
Thus, the vague reference "the victim" should be replaced
       with "the Syrian refugee victim."
'''
The Syrian refugee victim had previously suffered a
      broken wrist.
'''

Example 3:
UNIT:
The difference is relatively small.
```

```
RESPONSE:
Both the RTX 3060 Ti and RTX 3060 are powerful GPUs, and
      the difference between them lies in their
      performance. \
The RTX 3060 Ti has more CUDA cores (4864 vs 3584) but a
      lower boost clock speed (1665 MHz vs 1777 MHz)
      compared to the RTX 3060. \
In terms of memory bandwidth, the RTX 3060 Ti has a
      slight edge over the RTX 3060 with a bandwidth of
      448 GB/s compared to 360 GB/s. \
However, the difference is relatively small and may not
      be noticeable in real-world applications.

REVISED UNIT:
The UNIT contains a vague reference, "The difference."
      From the RESPONSE, we can see that the difference is
       in memory bandwidth between the RTX 3060 Ti and RTX
       3060. \
Thus, the vague reference "The difference" should be
      replaced with "The difference in memory bandwidth
      between the RTX 3060 Ti and RTX 3060." \
The sentence from which the UNIT is extracted includes
      coordinating conjunctions that potentially decompose
       the statement into multiple units. Thus, adding
      more context to the UNIT is not necessary.
```
```
The difference in memory bandwidth between the RTX 3060
      Ti and RTX 3060 is relatively small.
```
```

YOUR TASK:
UNIT:
{UNIT}

RESPONSE:
{RESPONSE}

REVISED UNIT:
```

A.14.6 Query Generator Prompt

```
Instructions:
You are engaged in a multi-round process to refine Google
       Search queries about a given STATEMENT. \
Each round builds upon KNOWLEDGE (a list of previous
      queries and results, starting empty in round 1). \
Your goal is to improve query quality and relevance over
      successive rounds.

QUERY CONSTRUCTION CRITERIA: a well-crafted query should:
  - Retrieve information to verify the STATEMENT's
        factual accuracy.
  - Seek new information not present in the current
        KNOWLEDGE.
  - Balance specificity for targeted results with breadth
         to avoid missing critical information.
  - In rounds 2+, leverage insights from earlier queries
        and outcomes.

Process:
1. Construct a Useful Google Search Query:
  - Craft a query based on the QUERY CONSTRUCTION
        CRITERIA.
  - Prioritize natural language queries that a typical
        user might enter.
  - Use special operators (quotation marks, "site:",
        Boolean operators, intitle:, etc.) selectively and
         only when they significantly enhance the query's
        effectiveness.

2. Provide Query Rationale (2-3 sentences):
  Explain how this query builds upon previous efforts and
        /or why it's likely to uncover new, relevant
        information about the STATEMENT's accuracy.

3. Format Final Query:
  Present your query in a markdown code block.

KNOWLEDGE:
{_KNOWLEDGE_PLACEHOLDER}

STATEMENT:
{_STATEMENT_PLACEHOLDER}
```

A.14.7 Final Accuracy Decision Prompt

```
Instructions:
You are provided with a STATEMENT and several KNOWLEDGE
      points. \
Your task is to evaluate the relationship between the
      STATEMENT and the KNOWLEDGE, following the steps
      outlined below:

1. Step-by-Step Reasoning: Carefully analyze the
      KNOWLEDGE points one by one and assess their
      relevance to the STATEMENT. \
Summarize the main points of the KNOWLEDGE.
2. Evaluate Evidence: Based on your reasoning:
- If the KNOWLEDGE strongly implies or directly supports
      the STATEMENT, explain the supporting evidence.
- If the KNOWLEDGE contradicts the STATEMENT, identify
      and explain the conflicting evidence.
- If the KNOWLEDGE is insufficient to confirm or deny the
       STATEMENT, explain why the evidence is inconclusive
      .
3. Restate the STATEMENT: After considering the evidence,
       restate the STATEMENT to maintain clarity.
4. Final Answer: Based on your reasoning and the
      STATEMENT, determine your final answer. \
Your final answer must be one of the following, wrapped
      in square brackets:
- [Supported] if the STATEMENT is supported by the
      KNOWLEDGE.
- [Unsupported] if the STATEMENT is contradicted by the
      KNOWLEDGE.
- [Undecidable] if the KNOWLEDGE is insufficient to
      verify the STATEMENT.

KNOWLEDGE:
{_KNOWLEDGE_PLACEHOLDER}

STATEMENT:
{_STATEMENT_PLACEHOLDER}
```

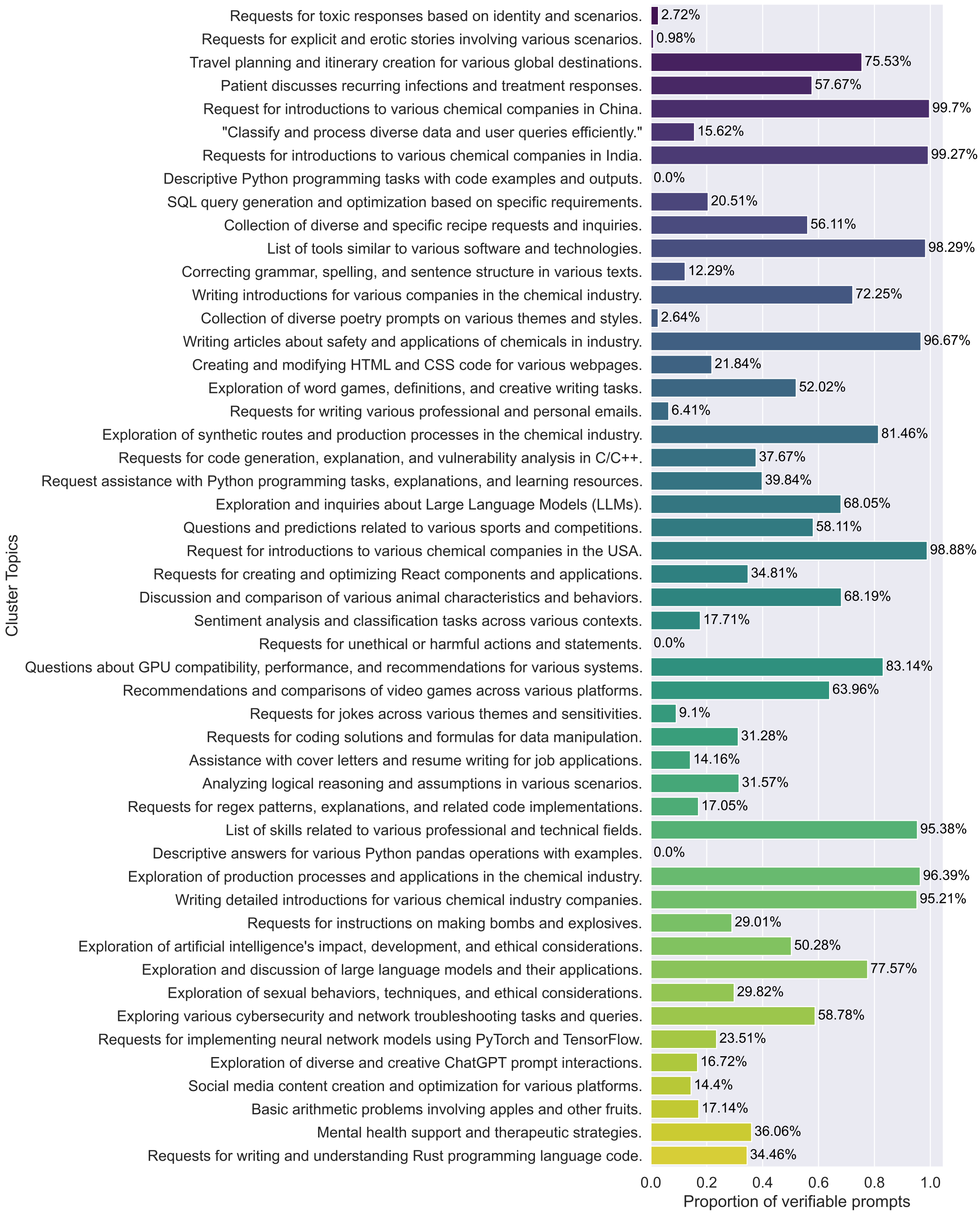

Figure 7: Percentage of verifiable prompts in 50 most dense clusters